**Documentación**

**Lote de Pruebas “*Pedregal*”**

**Programación Avanzada (2013)**

**Normalización de Datos de Entrada / Datos de Salida**

**Datos de Entrada**

Se recibe un archivo **<***nombre***>**.in que contiene:

1. **Primera línea:** las dimensiones Dx y Dy (1 ≤ Dx, Dy ≤ 1000) del terreno, separadas por un blanco.
2. **Segunda línea:** las dimensiones de la casa F y L (1 ≤ L ≤ F ≤ 100), separadas por un blanco.
3. **Tercera línea:** el número de peñascos P (1 ≤ P ≤ 1000).
4. **P-últimas líneas:** P-líneas conteniendo la fila y la columna de cada cuadrícula que contiene un peñasco, separadas por un blanco.

**Datos de Salida**

El programa debe generar un archivo **<***nombre***>**.out que contiene:

1. **Primera línea:**
   1. “NO”: en el caso que no sea posible construir la casa.
   2. “SI”: en el caso que sea posible construir la casa.

En este último caso, además:

1. **Segunda línea:** menor número de fila y columna del casillero donde está ubicada la casa, separados por blanco.
2. **Tercera línea:** indicando la orientación de la casa (“N”, “S”, “E”, “O”), de acuerdo a la orientación del la puerta de entrada.

|  |  |
| --- | --- |
| Caso de Prueba | 01 |
|  | |
| Nombre | 00\_CONSIGNA\_caso\_base.in |
| Tipo | Derivada de requisitos. |
| Descripción | Este caso de prueba es propuesto como inicial y debe validar correctamente. Como situación se tiene un pedregal de 9x6 con diez piedras y una casa de 3x2, ubicable en sentido Sur, en el punto (3,2) del pedregal. |
| Archivo de Entrada | *00\_CONSIGNA\_caso\_base.in* |
| Entrada | 9 6  3 2  10  1 1  1 6  2 1  2 3  5 1  5 5  6 2  7 3  7 5  9 1 |
| Archivo de Salida | *00\_CONSIGNA\_caso\_base.out* |
| Salida (esperada) | SI  3 2  S |
| NOTAS | **EXISTEN MÚLTIPLES SOLUCIONES.**  El archivo de salida es “propuesto” como una posible solución. Utilizar programa probador según criterio. |

|  |  |
| --- | --- |
| Caso de Prueba | 02 |
|  | |
| Nombre | 01\_caso\_minimalista. |
| Tipo | Derivada del análisis. |
| Descripción | Este caso de prueba es propuesto como caso minimalista para los límites inferiores de las restricciones impuestas por la consigna: la casa debe poder entrar en sus mínimas dimensiones (1x1), en el menor pedregal que admite dicha situación (2x1) con una piedra. |
| Archivo de Entrada | *01\_caso\_minimalista.in* |
| Entrada | 2 1  1 1  1  2 1 |
| Archivo de Salida | *01\_caso\_minimalista.out* |
| Salida (esperada) | SI  1 1  S |
| NOTAS | **EXISTEN MÚLTIPLES SOLUCIONES.**  El archivo de salida es “propuesto” como una posible solución. Utilizar programa probador según criterio. |

|  |  |
| --- | --- |
| Caso de Prueba | 03 |
|  | |
| Nombre | 02\_FATIGA\_caso\_maximal.in |
| Tipo | Derivada del análisis. |
| Descripción | Este caso de prueba es propuesto como caso maximal para los límites superiores de las restricciones impuestas por la consigna: la casa debe poder entrar en sus máximas dimensiones (100x100), en el mayor pedregal que admite dicha situación (1000x1000) con mil piedras. Notar que este caso también es de fatiga pues debe recorrer todo el pedregal probando la ubicación de la casa casi todas las veces hasta encontrar la piedra. |
| Archivo de Entrada | *02\_FATIGA\_caso\_maximal.in* |
| Entrada | Ver archivo (omitido por extensión) |
| Archivo de Salida | *02\_FATIGA\_caso\_maximal.out* |
| Salida (esperada) | NO |

|  |  |
| --- | --- |
| Caso de Prueba | 04 |
|  | |
| Nombre | 03\_caso\_girar\_casa.in |
| Tipo | Derivada de requisitos. |
| Descripción | Este caso de prueba propone demostrar que dada una casa de forma horizontal, el algoritmo de resolución sea capaz de ubicar la casa de forma vertical (único resultado posible). |
| Archivo de Entrada | *03\_caso\_girar\_casa.in* |
| Entrada | 4 3  3 2  4  1 1  1 3  4 1  4 3 |
| Archivo de Salida | *03\_caso\_girar\_casa.out* |
| Salida (esperada) | SI  2 1  E |

|  |  |
| --- | --- |
| Caso de Prueba | 05 |
|  | |
| Nombre | 04\_varios\_resultados.in |
| Tipo | Derivada de requisitos. |
| Descripción | Este caso de prueba propone demostrar que de existir más de una solución posible, obtengamos una de ellas en el archivo de salida. |
| Archivo de Entrada | *04\_varios\_resultados.in* |
| Entrada | 6 3  2 2  1  2 1 |
| Archivo de Salida | *04\_varios\_resultados.out* |
| Salida (esperada) | SI  1 2  O |

|  |  |
| --- | --- |
| Caso de Prueba | 06 |
|  | |
| Nombre | 05\_casa\_mayor\_a\_terreno.in |
| Tipo | Derivada del análisis. |
| Descripción | Este caso de prueba pretende demostrar que de darse una casa de mayores dimensiones que el terreno, no obtengamos resultado posible. |
| Archivo de Entrada | *05\_casa\_mayor\_a\_terreno.in* |
| Entrada | 2 2  3 3  1  1 1 |
| Archivo de Salida | *05\_casa\_mayor\_a\_terreno.out* |
| Salida (esperada) | NO |

|  |  |
| --- | --- |
| Caso de Prueba | 07 |
|  | |
| Nombre | 06\_terreno\_lleno\_de\_peñascos.in |
| Tipo | Derivada de requisitos. |
| Descripción | Este caso de pruebas demuestra que no puede ubicar la casa si es que el terreno está lleno de peñascos. |
| Archivo de Entrada | *06\_terreno\_lleno\_de\_peñascos.in* |
| Entrada | 3 2  2 2  6  1 1  2 1  3 1  1 2  2 2  3 2 |
| Archivo de Salida | *06\_terreno\_lleno\_de\_peñascos.out* |
| Salida (esperada) | NO |